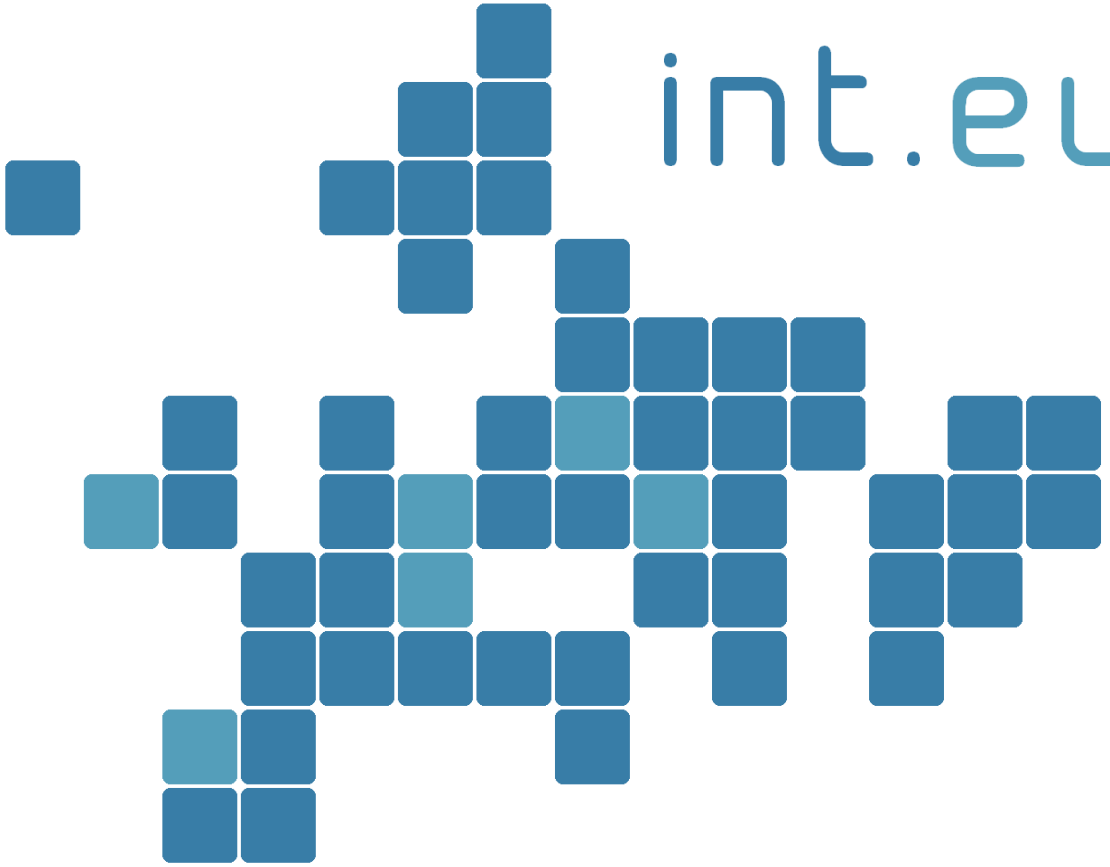


int.eu.grid

http://www.interactive-grid.eu

Grids

Marcus Hardt
Forschungszentrum Karlsruhe

Outline

Part I

- Currently existing grid-middlewares
  - Unicore
  - Globus
  - gLite
    - Middleware
    - Infrastructure

Part II

- Interactive European Grid
  - Interactivity
  - GridSolve
  - Grid + Matlab

Unicore

- Lead by Juelich Supercomputer Centre
- Goals
  - Supercomputer centers support users
  - Abstraction of heterogenous resources
  - Ease of use + administration
  - Security x.509
- Funding by BMBF + EU
- Target Infrastructure:
  - Supercomputers
    => **Low number** of **high power** resources
  - e.g.: JUMP (IBMp690), JUBL(Blue Gene/L)

Unicore

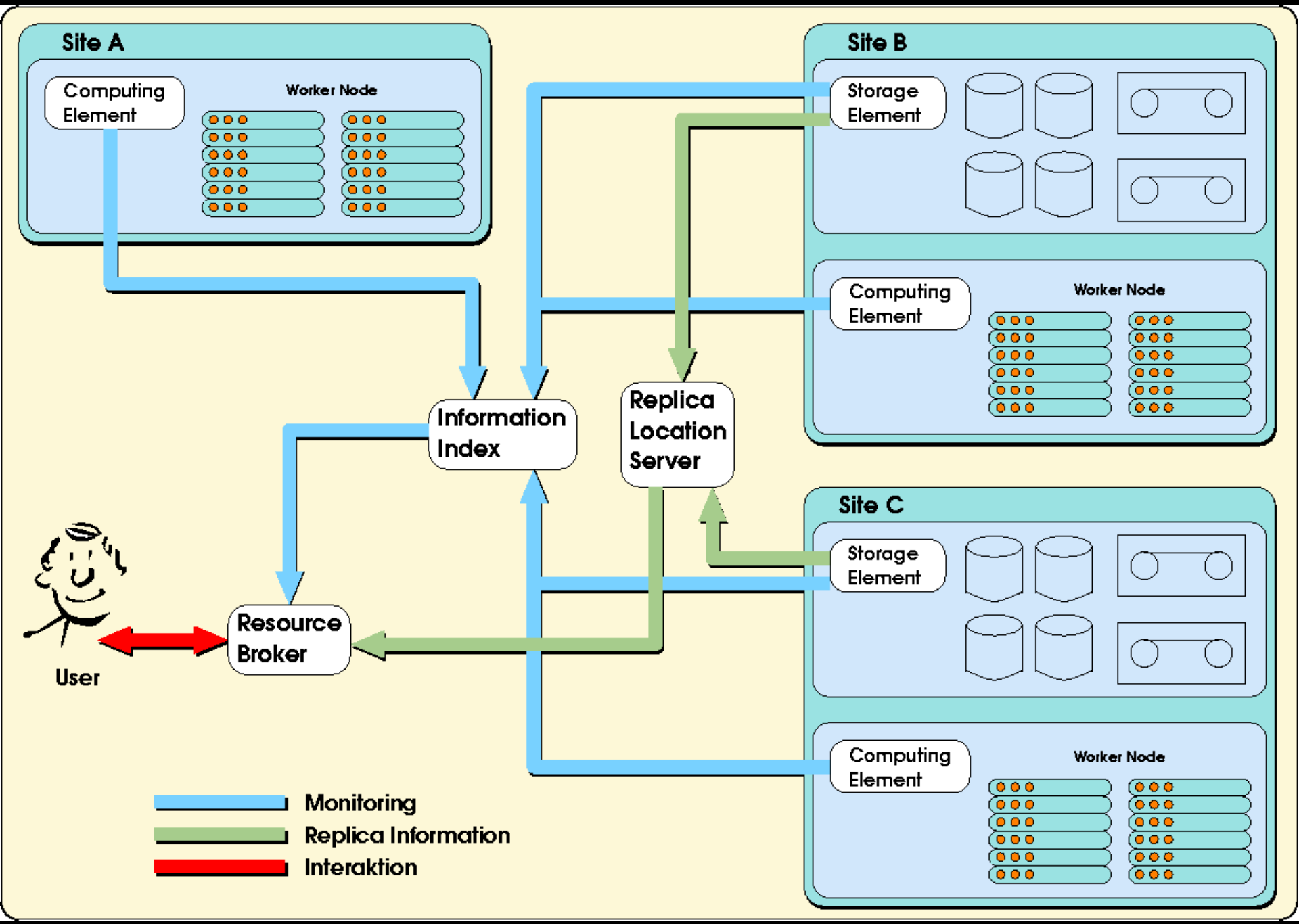
- Lead by Juelich Supercomputer Centre
- Goals
  - Supercomputer centers support users
  - Abstraction of heterogenous resources
  - Ease of use + administration
  - Security x.509
- Funding by BMBF + EU
- Target Infrastructure:

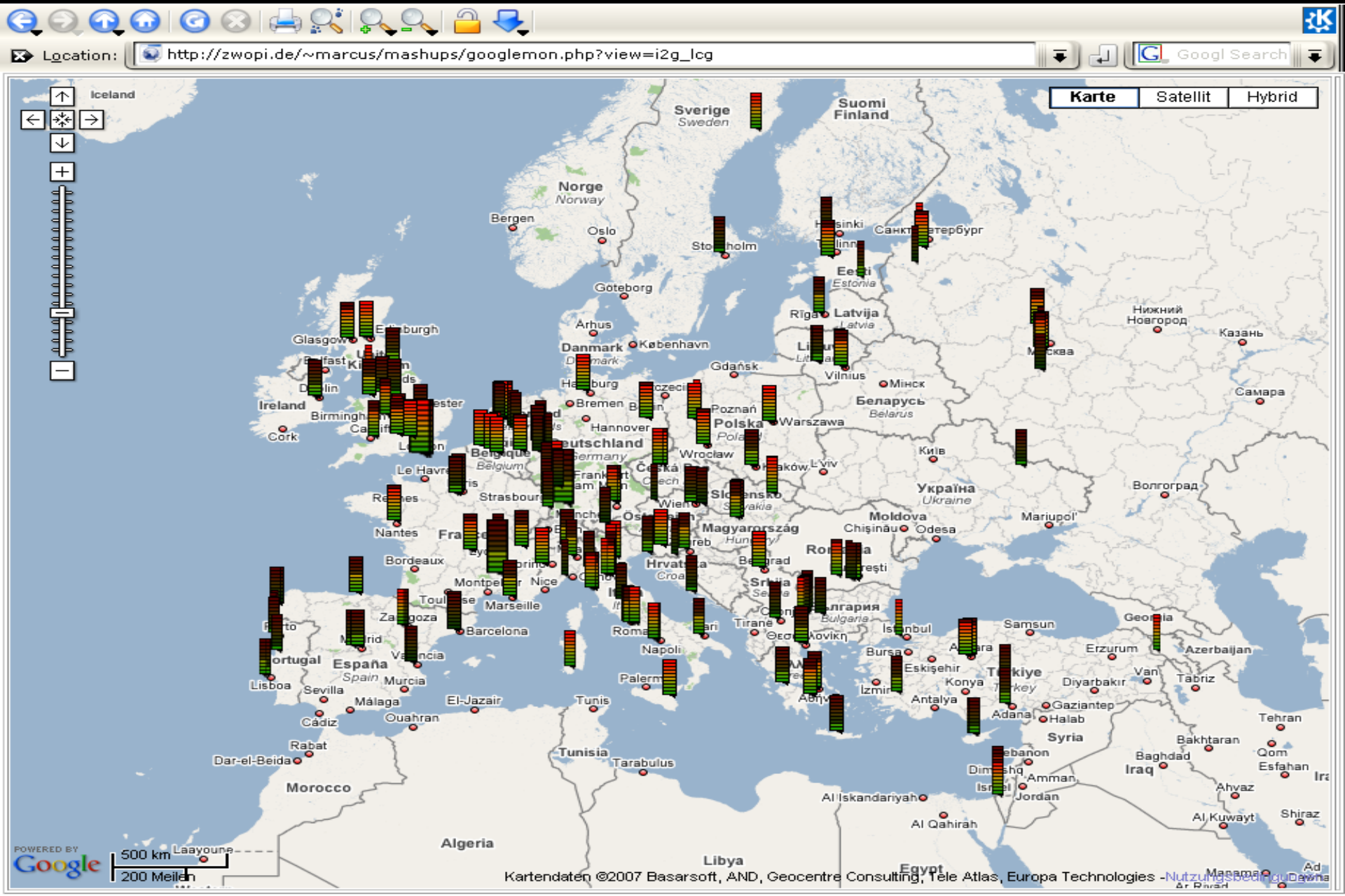
Globus

- Ian Foster (ANL) + Carl Kesselman (ISI)
- Goal:
  - Toolkit for distributed computing
- Tools
  - Authentication (X.509)
  - Monitoring (MDS)
  - Job submission (GRAM)
  - Data transfers (gridFTP)
  - Replication of Data (RLS)
  - ...
- Version 2: Job based
- Version 4: service oriented Web services based

gLite (=EDG, LCG, egee)

- CERN
  - Tools for data+computing of new accelerator
  - 10TB/year * 20years, random access
- Historical
  - Build on top of globus-2
  - Replace/improve parts
- Paradigm: **Send job to where the data is**
- Building blocks integrate functionality


Site A
Computing Element
Worker Node
Site B
Storage Element
Computing Element
Worker Node
Information Index
Replica Location Server
Site C
Storage Element
Computing Element
Worker Node
Resource Broker
User
Monitoring
Replica Information
Interaktion



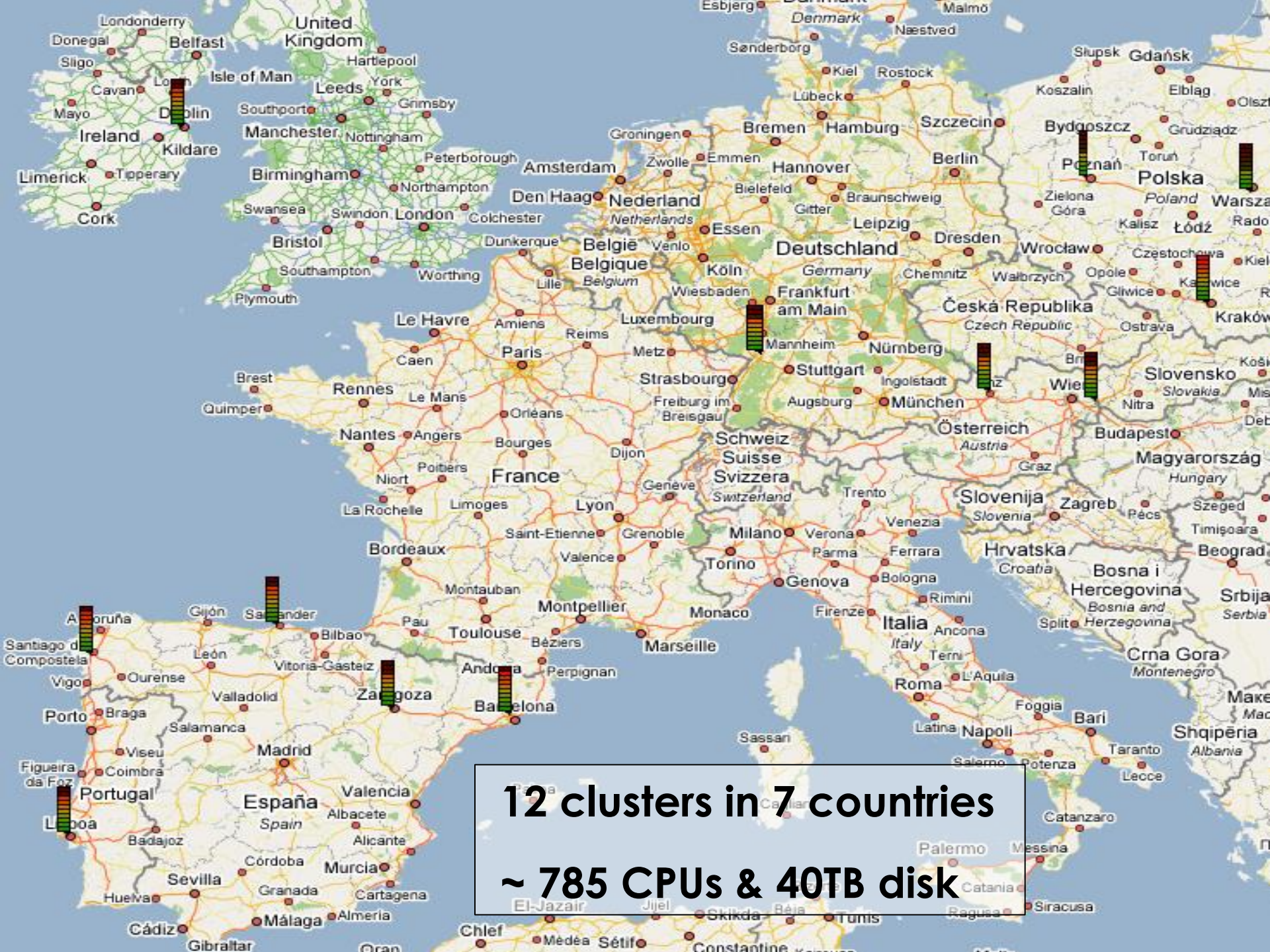
- Sites: 243 (in 49 countries)
- CPUS: 42798 (176 per site)
- RAM: 19TB
- RAM/CPU: 468MB
- DISK [Tot / Avail]: [8042TB / 5408TB] ([33892GB / 22792GB] per site)

One grid?

- 72 grid-related EU-projects
  - (search grid+IST at cordis.europa.eu)
- Why smaller grids?
  - Own the infrastructure
  - Know their owners
  - Influence on development
- FZK involvement:
  - EGEE
  - DGRID
  - **Interactive European Grid Project**
    = **int.eu.grid**
    = **i2g**

interactive grid

- 2 Year Project (May'06 - April'08)
- ~20 people
- Mission
  - 100% gLite compatible
  - MPI for the grid
  - Bring grid to new user communities
  - Improve useability
- Application areas
  - Fusion
  - Medicine (USCT)
  - Environment
  - Astrophysics

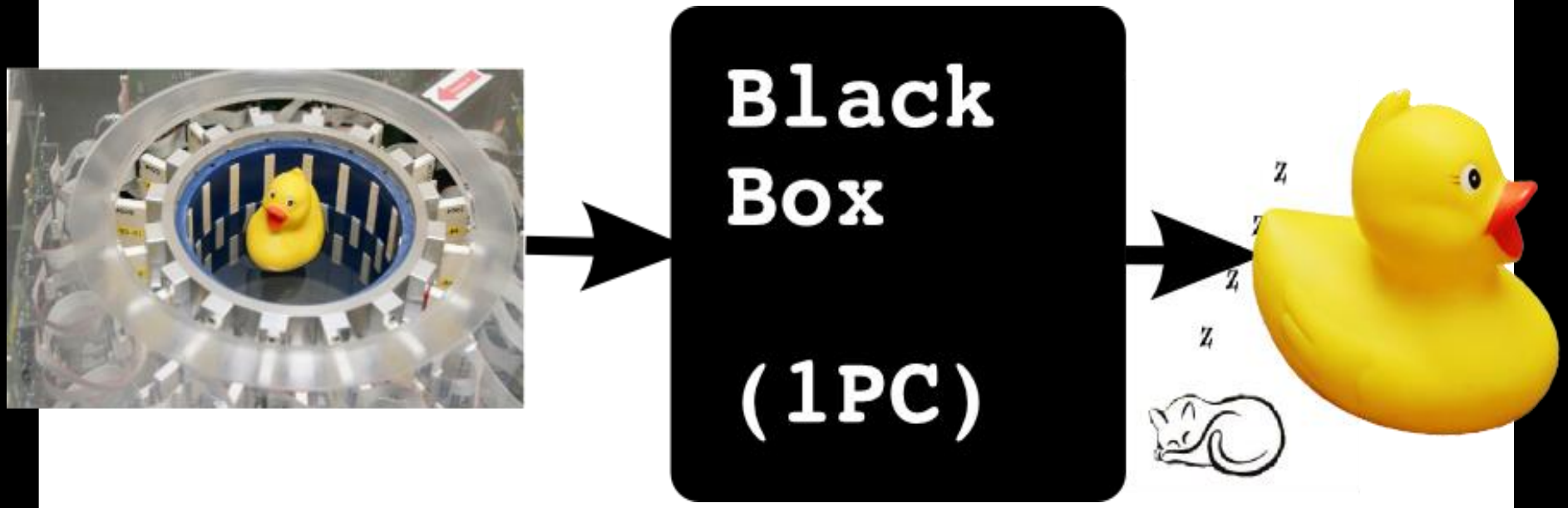

12 clusters in 7 countries
~ 785 CPUs & 40TB disk

What is interactivity?

- 1. Avoid queuing of important jobs
- 2. Online job steering
- 3. Interactive stdin / stdout transport
- 4. Direct (network) access to compute resources

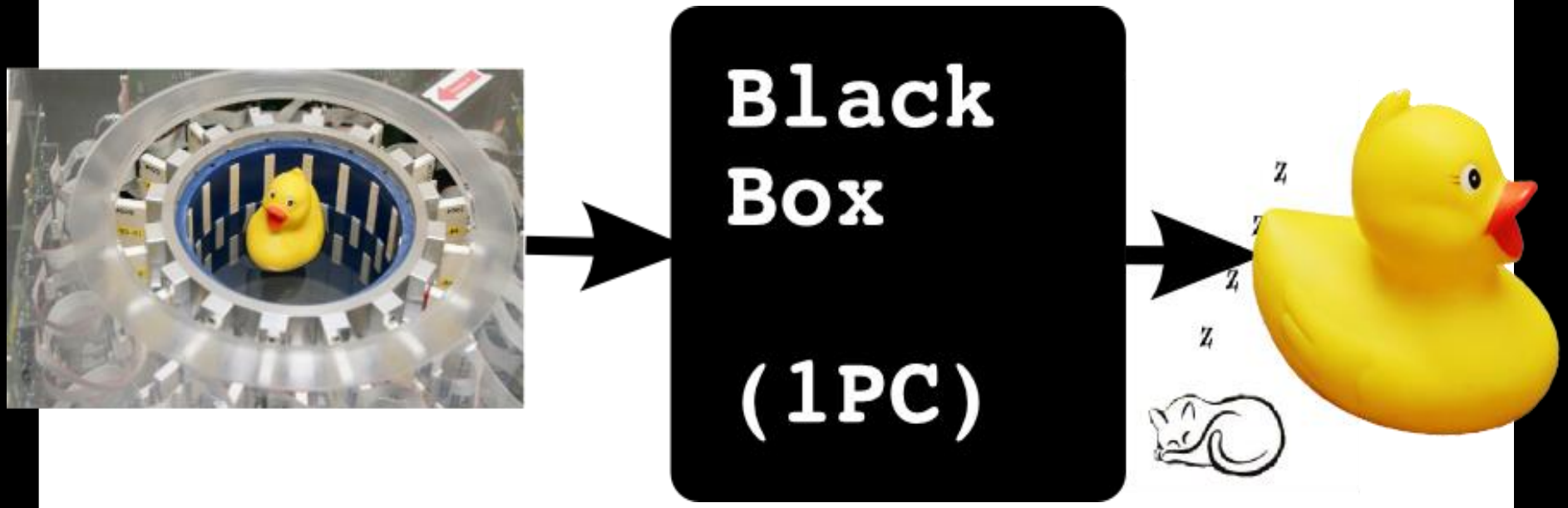
  => This is my task

The USCT application at FZK



- Computation takes long (days, weeks, years)

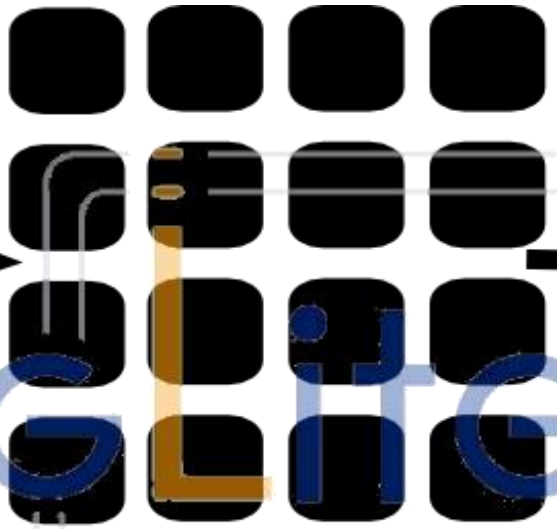
Typical problem

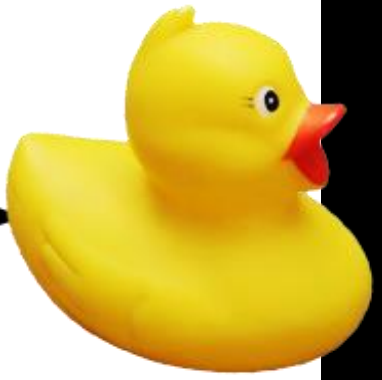


- Computation takes long (days, weeks, years)
- **Goal:**
  - **Seamless, interactive, grid access**
  - **easy enough for scientists**

Using gLite

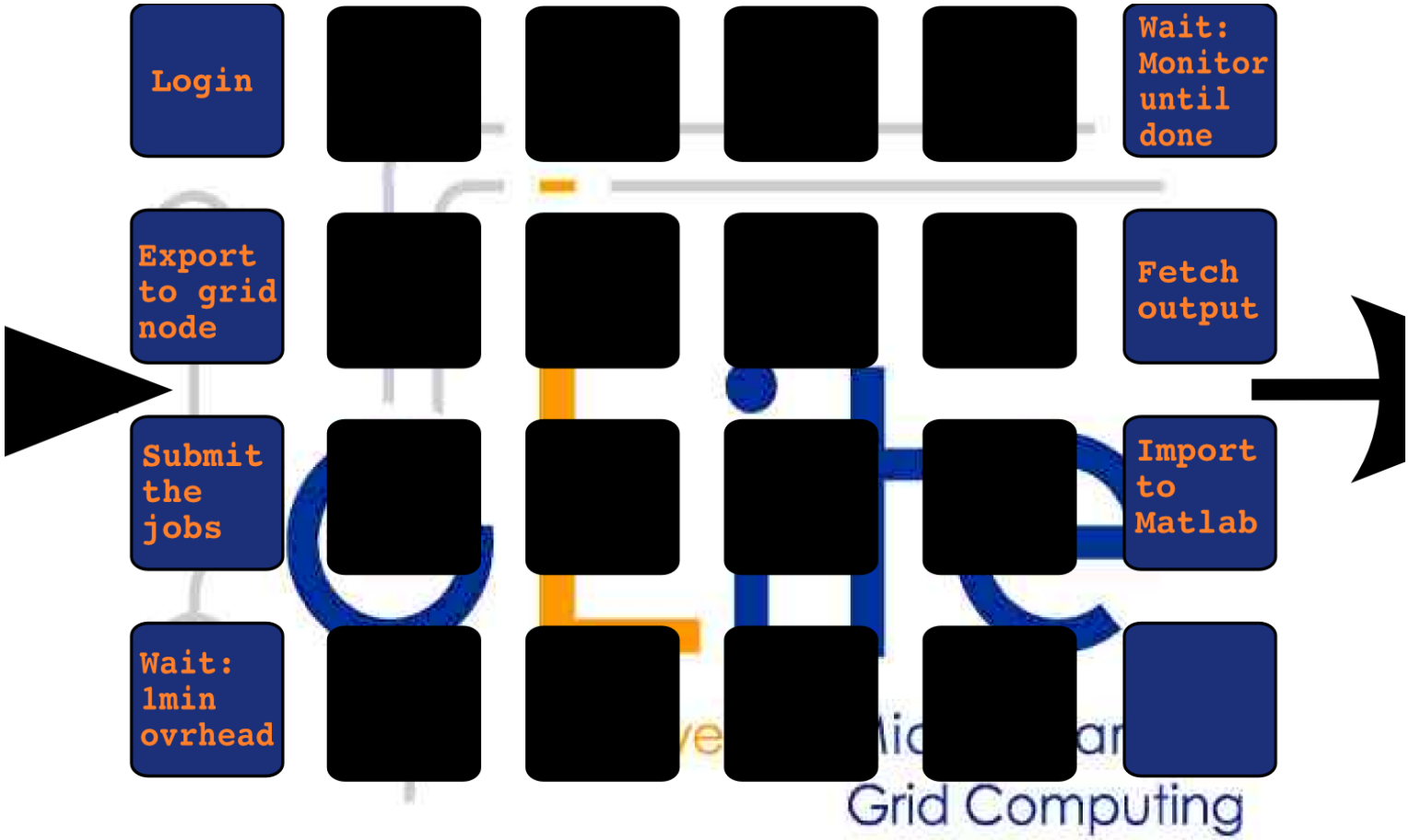
- Initial approach to parallel execution:
  - Partitioning of data
  - Many parallel jobs





Using gLite

- Lets take a closer look



Using gLite

- Goal:
  - Seamless
  - Interactive
  - Grid access
  - For scientists

What's missing?

- **Goal:**
  - **Seamless**
  - **Interactive**
  - **Grid access**
  - **For scientists**

- Seamless
  - User might not know if he uses the grid
- Interactive
  - No overhead (< 10 s)
  - No manual data movement
- From Matlab
  - Run Matlab-functions remotely

Improving Grid Access with GridSolve

- GridSolve
  - Developed at ICL, University Tennessee, Knoxville
  - Implements an client/agent/server solution
  - Client interface for Java, C, Fortran, **Matlab**, Octave
  - Easy to use:

    y=problem(x) <=> y=gs_call('problem', x)

    - Transport input parameters to remote side
    - Execute "problem"
    - Transport result back

=> Reduce complexity of the grid to one function call

How to do it?

- **Goal:**
  - **Seamless**
  - **Interactive**
  - **Grid access**
  - **For scientists**

- 1. Integrate GridSolve with gLite
- 2. Make Matlab run on gLite

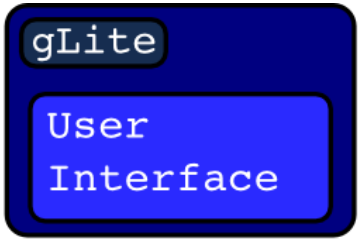
=> **G**rid **i**n **M**atlab using **G**ridsolv**e** & **R**PC
GIMGER (speak: ginger)

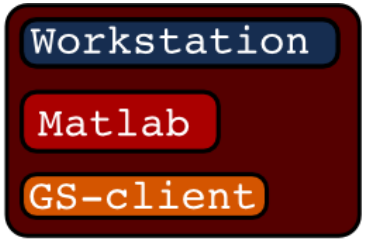
GridSolve(GS)/gLite Integration

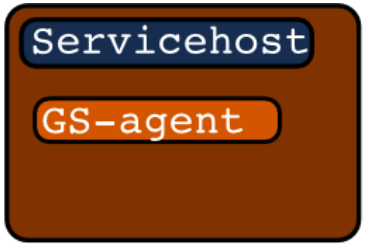
- Create GS-Service hosts (GS-agent)
- Send 100s of GS-servers to gLite infrastructure
  - Setup build infrastructure
  - Package GridSolve + Matlab Runtime
  - Create gLite jobs
  - Install (GS+ML) on Workernodes
- Ensure network connectivity
  - GS-client, GS-agent, GS-Proxy, GS-Server

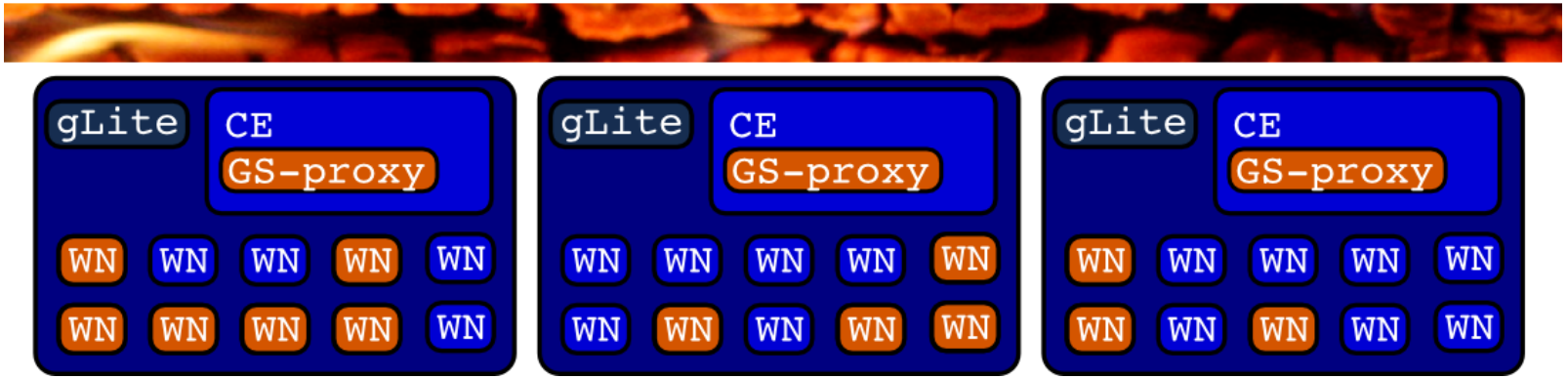
GridSolve startup on gLite

GridSolve ready for action

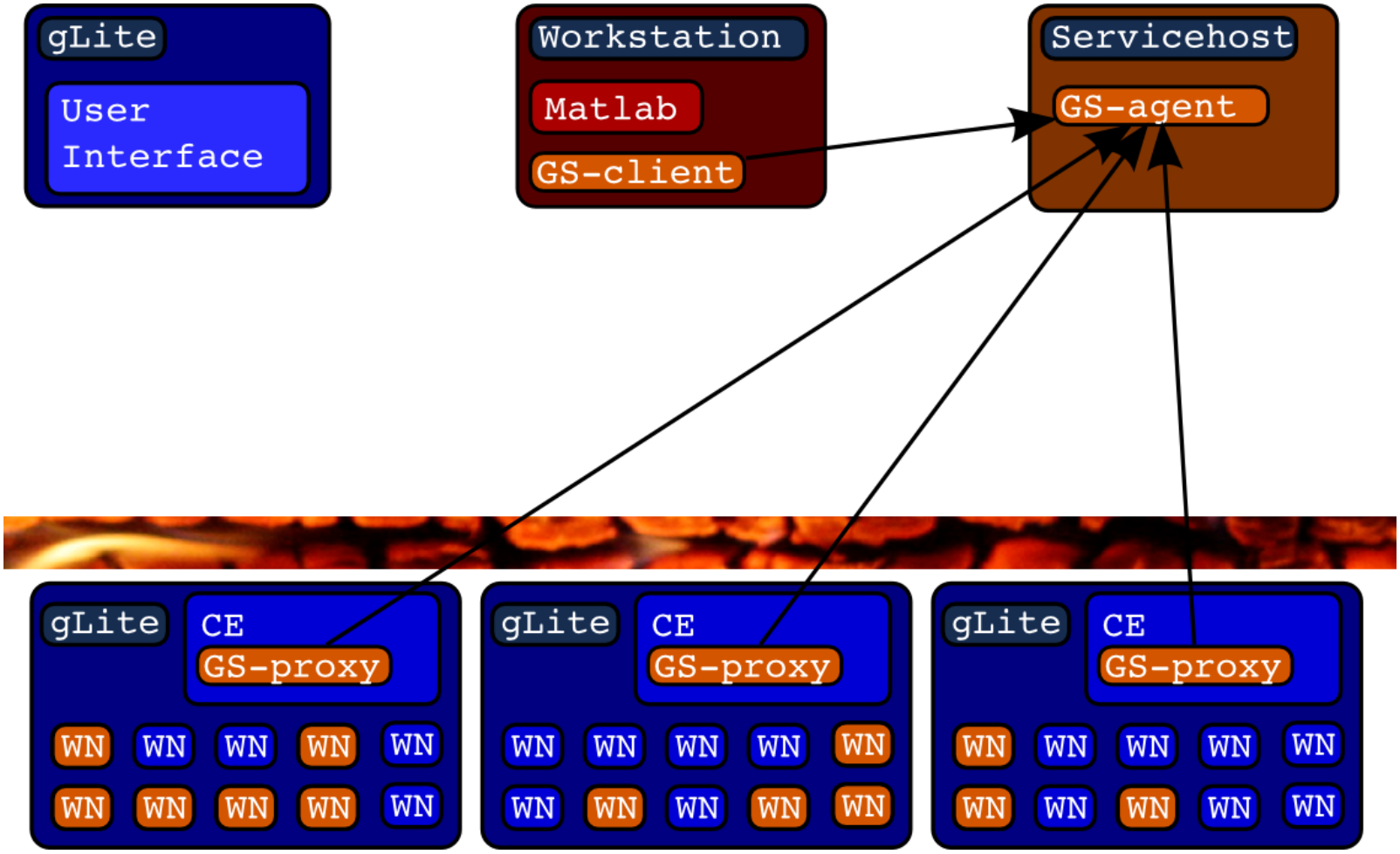




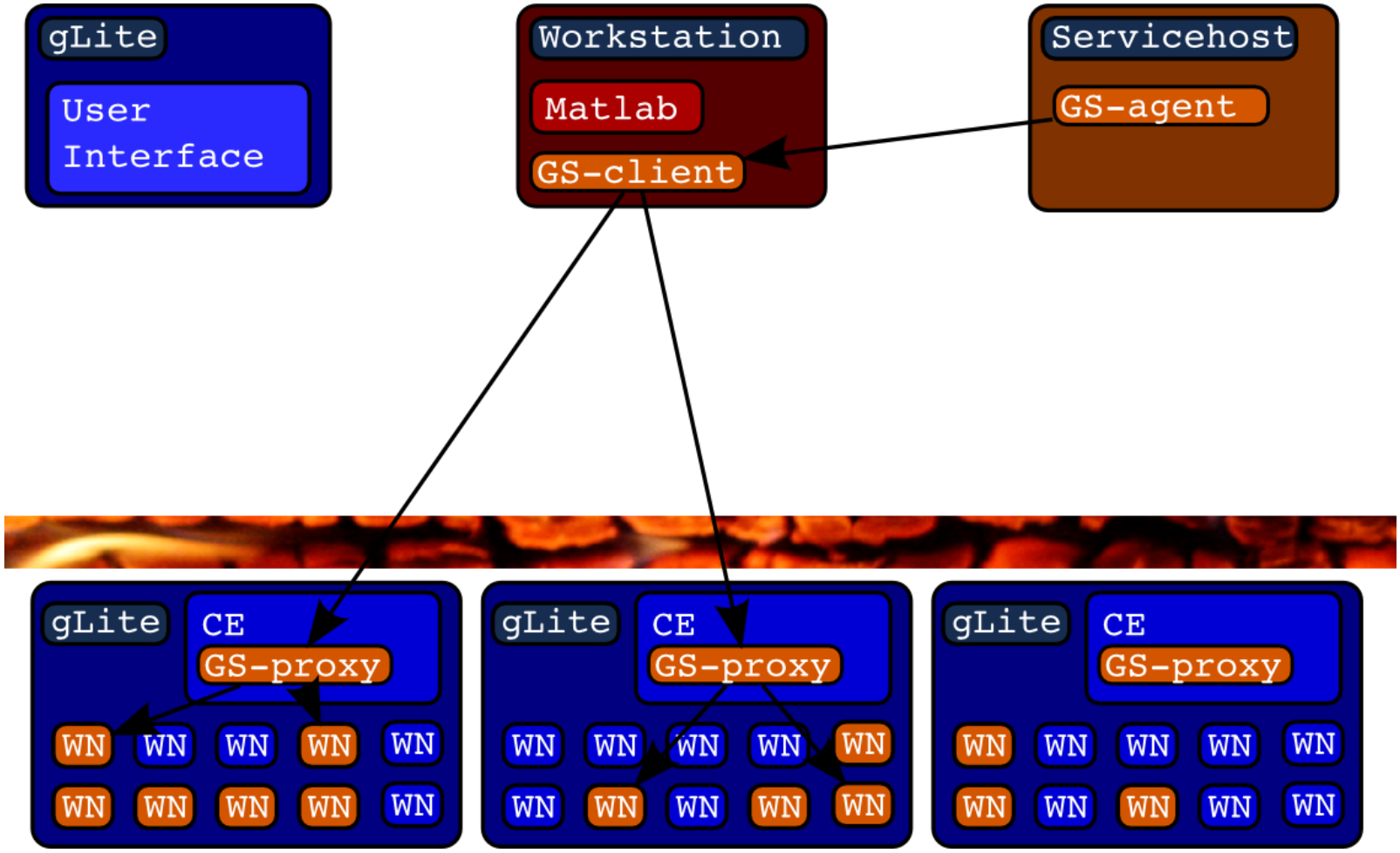




GridSolve ready in action

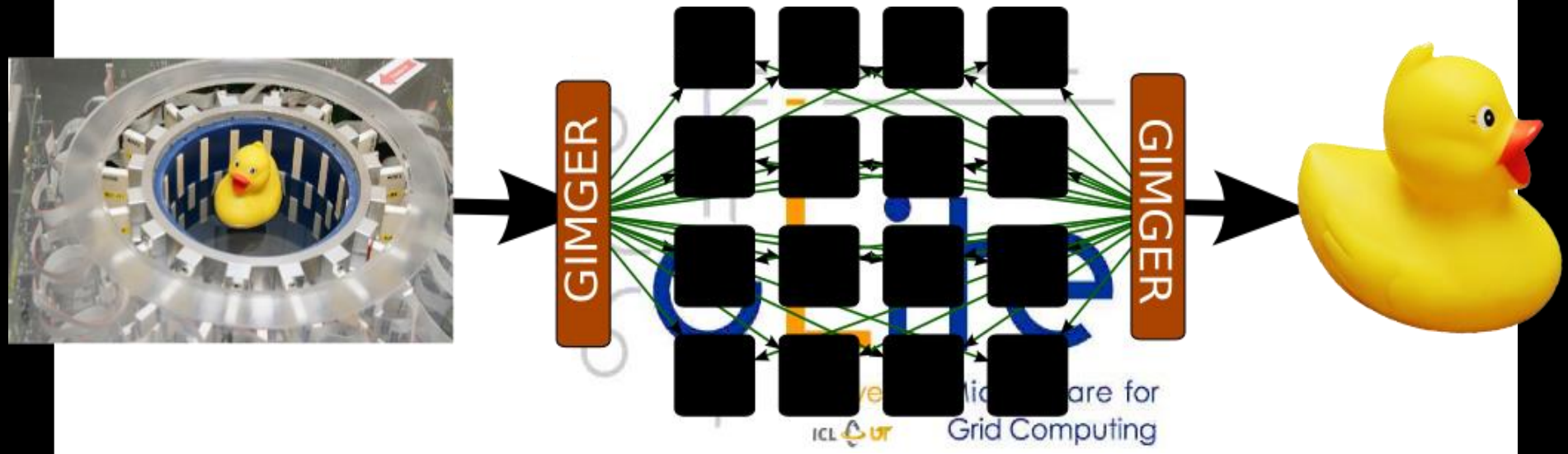


GridSolve ready in action



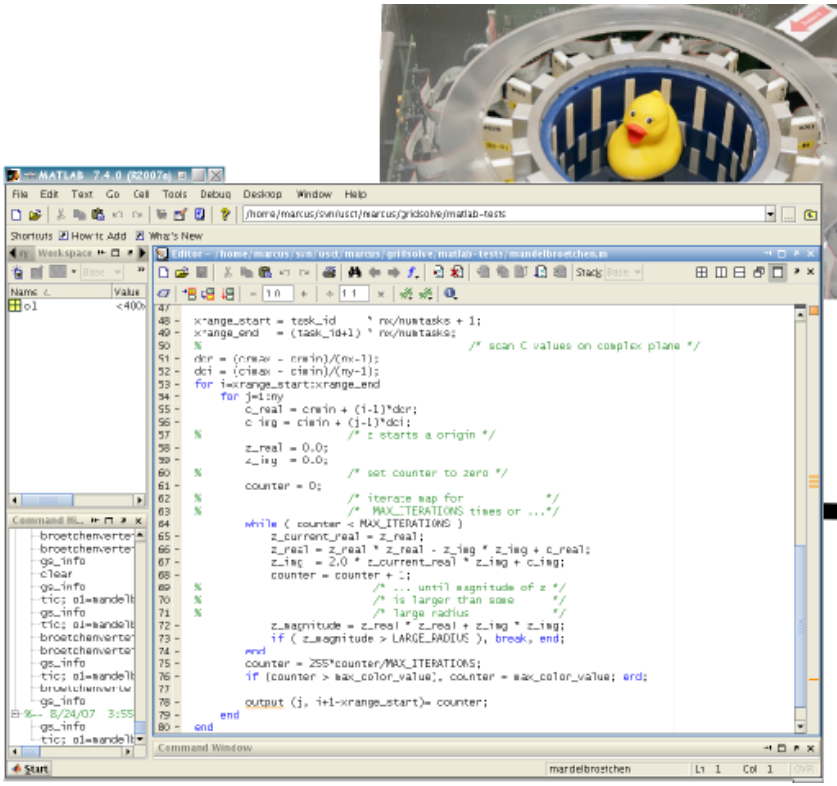
Putting things together

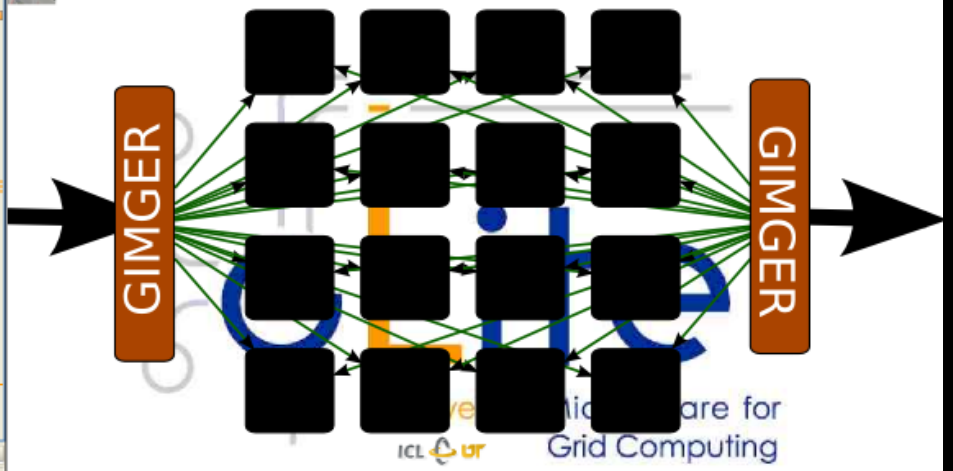
- RPC with GridSolve
- On top of int.eu.grid/gLite
- Using Matlab functionality
- GIMGER



Demonstration

- Simulation: Mandelbrot fractal
- Using the same infrastructure



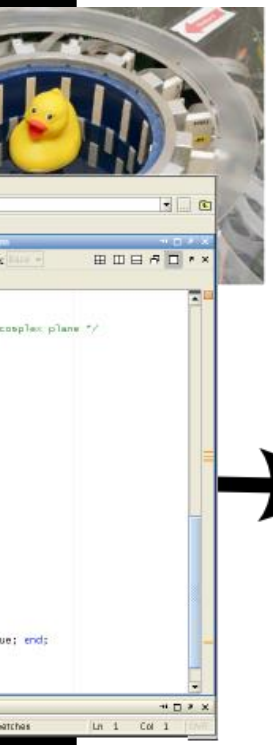


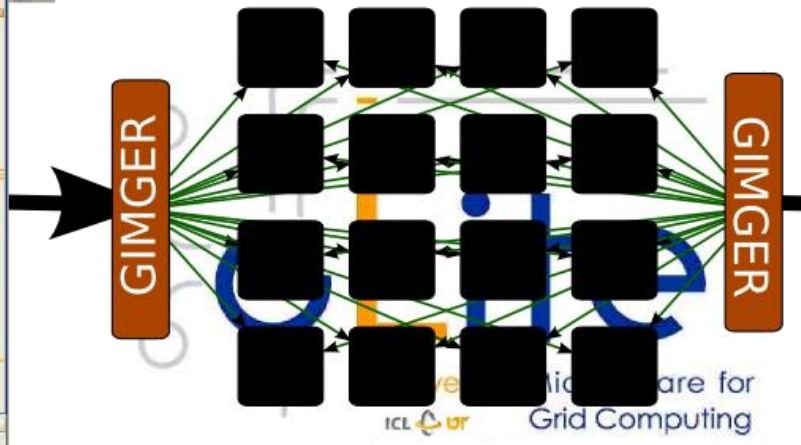
Life-Demo

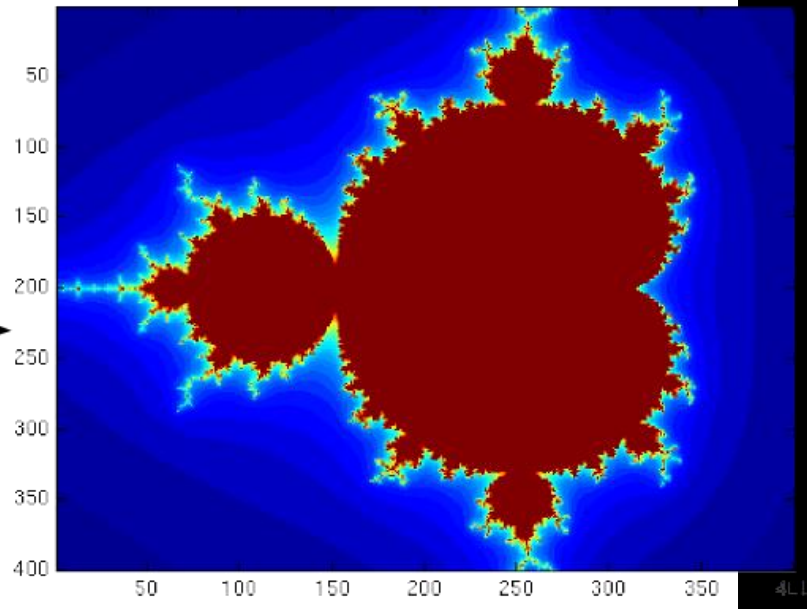
- Movie of the life demonstration:
  - **http://marcus.hardt-it.de/grid4matlab**
- **Real** life demo on int.eu.grid

Result

- Simulation works
- Reasonable speedup (4x on 8 machines)







Summary

- We can
  - Convert Matlab functions to run on the grid
    - Involves hands-on work
  - Run simple simulations in our infrastructure
  - Use the grid from matlab...
    ... for hand-tuned functions
- We want to...
  - Use real code
    - Cope with the data (20 GB in, 8 GB out)
      - Use gLite data handling methods
    - Identify Bottlenecks
  - Automatically send Matlab functions to the grid
    - Reduce hands-on work


Google
interactive grid
I'm Feeling Lucky